

A Robust High-Order Discontinuous Galerkin Solver for Fluid Flow with Cavitation

Malte Hoffmann*, Sebastian Boblest**, Philipp Offenhäuser†, and Claus-Dieter Munz*
Corresponding author: hoffmann@iag.uni-stuttgart.de

* Institute of Aerodynamics and Gas Dynamics, Pfaffenwaldring 21

** Visualization Research Center, Allmandring 19

† High Performance Computing Center, Nobelstrasse 19
University of Stuttgart, 70569 Stuttgart, Germany

Abstract: We present a computational fluid dynamics framework for the efficient and accurate simulation of multiphase flows. The compressible Navier-Stokes equations are solved with a Discontinuous Galerkin Spectral Element Method (DG-SEM). The system of governing equations is closed by the Helmholtz energy function, a highly accurate real multiparameter equation of state (EOS). This EOS covers liquid, vapor, liquid-vapor and supercritical regions. Thermodynamic equilibrium is assumed in the multiphase region and a dense gas approach is used to represent both phases in the solver. The EOS is evaluated and stored in a table during a parallelized preprocessing step, which we discuss in detail. A strategy for efficiently building the needed tables during calculation and post-processing is presented. During phase-change, high gradients are possible, which cannot be handled by the DG-SEM. To handle these, a 2nd order finite-volume shock-capturing technique is used. 1D validation calculations are shown and also 2D and 3D use-case calculations on up to 6.000 cores.

Keywords: high-order, compressible Navier-Stokes, highly accurate equation of state, efficient table approach, FV shockcapturing, high-performance-computing, cavitation.

1 Introduction

During the last decades the perfect gas assumption was commonly used to close the compressible Navier-Stokes equation. In recent years, the focus has been moving to more realistic equations of state (EOS), because of the need to solve more complex problems like high-pressure injectors or rocket engines. In these environments, also phase transition can occur, e.g., in cavitation or condensation. To resolve such phenomena, for a pure fluid at least two phases have to be taken into account, namely vapor and liquid. It is a common practice to resolve the phase interface by using a sharp or diffuse interface tracking [16, 22, 9, 6]. But these use stiffened gas EOS and the effects of phase transition are modeled in a second step and added at the interface position as source or jump terms. In this work however, to resolve such multiphase phenomena, a thermodynamic equilibrium assumption is considered in the multi-phase region of the EOS, which uses the Helmholtz energy functions. This means that the fraction of each phase depends only on density and temperature. This is also called the dense-gas approach with thermodynamic equilibrium. Similar approaches are also used by others [10, 11]. To solve the compressible Navier-Stokes equation, a Discontinuous Galerkin Spectral Element Method (DG-SEM) is used. The DG method was first introduced by Reed and Hill [15] in 1973 and later extended to non-linear conservation laws [8] like compressible gas dynamics. The first to solve the compressible Navier-Stokes equations with a DG method were Bassi and Rebay [3] in 2002. The high parallel efficiency of the DG-SEM, one of the many variants of DG methods, is shown in [1, 13]. The DG-SEM is limited to hexahedron elements but with curved faces, which allows complex unstructured meshes for calculation. A high-order method like this has the problem that shock

waves or strong gradients can lead to instabilities. The use of the highly accurate EOS leads also to high gradients during cavitation or condensation. It is mandatory to equip the DG-SEM solver with some sort of shock capturing. One option is to use explicit artificial viscosity. Persson and Peraire introduced this to DG methods [14] in 2006 and added viscosity locally. A second approach to capture shocks in a DG solver is hp-adaption. The DG cell is locally refined and a 2nd order finite volume method is used in the refined cells. The order is reduced but the resolution is increased. Since it needs to be thermodynamically consistent, the latter is used in this work. The complete solver is parallelized with MPI and performs very well on the CRAY XC40 (Hazel Hen) of the High Performance Computing Center Stuttgart (HLRS).

This paper is ordered like the following. In the second section the governing equation and the numerics is explained. Also Riemann solvers for real fluids and the shock-capturing technique are discussed. In the third section the efficient coupling of the EOS and the CFD-Solver is carried out. A very efficient table approach is used for this task. The fourth section contains 1D verification and 2D and 3D use-cases. The last section concludes the work and gives an outlook for future work.

2 Numerics

In this section a short overview about the DG-SEM and also the compressible Navier-Stokes equation is given. The used EOS is explained in more detail. Also some Riemann solvers for real fluid EOS are shown, the coupling of the DG-SEM and FV method is described and the used explicit time integration is explained. At the end of this section some information about the parallelization technique is given.

2.1 Navier-Stokes Equation

The compressible Navier-Stokes Equation can be written as

$$\frac{\partial U}{\partial t} + \nabla_x \cdot \vec{F}^a(U) - \nabla_x \cdot \vec{F}^d(U, \nabla_x U) = P, \quad (1)$$

where U is the vector of conservative variables $U = (\rho, \rho \vec{v}, \rho E)^T$, where ρ is the density, \vec{v} is the velocity vector in all three space dimensions and E is the total energy. $P \in \mathbb{R}^5$ is the source term vector, $\vec{F}^a = (F_1^a, F_2^a, F_3^a)^T$ are the inviscid or advection fluxes and $\vec{F}^d = (F_1^d, F_2^d, F_3^d)^T$ are the viscous or diffusion fluxes in three dimensional space. For Newtonian fluids the vector of the advection fluxes is given by

$$F_l^a(U) = \begin{pmatrix} \rho v_l \\ \rho v_1 v_l + \delta_{1l} p \\ \rho v_2 v_l + \delta_{2l} p \\ \rho v_3 v_l + \delta_{3l} p \\ \rho E v_l + p v_l \end{pmatrix}, \quad l = 1, 2, 3, \quad (2)$$

and the viscous fluxes by

$$F_l^d(U, \vec{\nabla} U) = \begin{pmatrix} 0 \\ \tau_{1l} \\ \tau_{2l} \\ \tau_{3l} \\ \tau_{lj} v_j - q_l \end{pmatrix}, \quad l = 1, 2, 3. \quad (3)$$

The viscous stress tensor is given as (with the viscosity μ)

$$\underline{\tau} := \mu(\nabla_x \vec{v} + (\nabla_x \vec{v})^T) - \frac{2}{3}(\nabla_x \cdot \vec{v})\underline{I}, \quad (4)$$

and $\vec{q} = (q_1, q_2, q_3)^T$ is the heat flux where λ is the heat conductivity and the flux is proportional to the temperature gradient,

$$\vec{q} := -\lambda \vec{\nabla} T. \quad (5)$$

2.2 Discontinuous Galerkin Spectral Element Method

Now the Navier-Stokes equation is used to derive the DG-SEM in 3D,

$$\begin{aligned}\frac{\partial U}{\partial t} + \nabla_x \cdot \vec{F}^a(U) - \nabla_x \cdot \vec{F}^d(U, \nabla_x U) &= 0, \\ \frac{\partial U}{\partial t} + \nabla_x \cdot \vec{F}(U, \nabla_x U) &= 0.\end{aligned}\tag{6}$$

Equation (6) is written in physical space, every element is mapped to a reference element $E \in [-1, 1]^3$ using the reference coordinates $(\xi^1, \xi^2, \xi^3)^T$. A polynomial mapping function $\vec{x}(\vec{\xi})$ is used to map from the reference element to the one in physical space. The Jacobian $J(\vec{\xi}) = \det(\frac{\partial \vec{x}}{\partial \vec{\xi}})$ is calculated from this mapping function. Transforming eq. (6) into reference space yields

$$U_t + \frac{1}{J(\vec{\xi})} \nabla_\xi \cdot \vec{\mathcal{F}}(U, \nabla_x U) = U_t + \frac{1}{J(\vec{\xi})} \nabla_\xi \cdot (\vec{\mathcal{G}}(U) - \vec{\mathcal{H}}(U, \nabla_x U)) = 0,\tag{7}$$

where $\nabla_\xi = (\partial_{\xi^1} \ \partial_{\xi^2} \ \partial_{\xi^3})^T$ is the divergence operator in reference space. The flux is transformed in $\vec{\mathcal{F}}(U, \nabla_x U)$ and the contributions of inviscid and viscous to the flux are the terms $\vec{\mathcal{G}}(U)$ and $\vec{\mathcal{H}}(U, \nabla_x U)$, respectively. In each element the solution vector, depending on the governing equation, is approximated by a tensor product of 1-D Lagrange polynomials ℓ^N of degree N ,

$$U(\vec{\xi}, t) \approx \sum_{i,j,k=0}^N \hat{U}_{ijk}(t) \psi_{ijk}^N(\vec{\xi}), \quad \psi_{ijk}^N(\vec{\xi}) = \ell_i^N(\xi^1) \ell_j^N(\xi^2) \ell_k^N(\xi^3).\tag{8}$$

This is a nodal interpolation ansatz and $\hat{U}_{ijk}(t)$ are time dependent nodal degrees of freedom (DOF). $\ell_i^N(\xi)$ stands for the standart Lagrange polynomial of degree N and is defined by a set of nodal points $\hat{U}_{ijk}(t)$ which can be defined arbitrarily. In this case, following Hindenlang et al. [13], the $N + 1$ Gauss-Legendre points are used as the interpolation node set. The same holds for the flux $\vec{\mathcal{F}}$

$$\mathcal{F}^l(\vec{\xi}) \approx \sum_{i,j,k=0}^N \hat{\mathcal{F}}_{ijk}^l \psi_{ijk}^N(\vec{\xi}), \quad l = 1, 2, 3\tag{9}$$

$$\hat{\mathcal{F}}_{ijk}^l = \mathcal{G}^l(U) - \mathcal{H}^l(U, \vec{\nabla}_x U) |_{\vec{\xi}_{ijk}}\tag{10}$$

Multiplying eq. (7) by a test function $\phi(\vec{\xi})$, which is from the same space as the polynomial interpolation basis function, and integrating over the reference element E , the variational formulation in reference space is obtained,

$$\int_E \left(J U_t + \nabla_\xi \cdot \vec{\mathcal{F}}(U, \nabla_x U) \right) \phi(\vec{\xi}) d\vec{\xi} = 0.\tag{11}$$

The differentiability requirement for the flux term can be moved to the test function by integration by parts. Allowing the solution to be discontinuous across element interfaces, the weak formulation is achieved

$$\int_E J U_t \phi d\vec{\xi} + \oint_{\partial E} \underbrace{(\mathcal{G}_n^* - \mathcal{H}_n^*)}_{\mathcal{F}_n^*} \phi ds - \int_E \vec{\mathcal{F}}(U, \nabla_x U) \cdot \nabla_\xi \phi d\vec{\xi} = 0,\tag{12}$$

where $\mathcal{G}_n^* := \mathcal{G}_n^*(U^+, U^-)$ is the inviscid numerical flux function (see subsection 2.4 for more detail) normal to the surface. The superscripts \pm stands for the solution at the grid cell surface from the neighbor grid cell and the local one, respectively.

To derive the viscous flux terms, the governing equations is rewritten with an additional variable \vec{S} as

an estimation to the solution gradients to a corresponding system of first order equations:

$$\begin{aligned}\vec{S} - \nabla_x U &= 0, \\ U_t + \nabla_x \cdot \vec{F}(U, \vec{S}) &= 0.\end{aligned}\tag{13}$$

Following the steps from above, the equation system leads to

$$\begin{aligned}\int_E J \vec{S} \phi d\vec{\xi} + \oint_{\partial E} \vec{\mathfrak{U}}_n^* \phi ds - \int_E U \cdot \nabla_\xi \phi d\vec{\xi} &= 0, \\ \int_E J U_t \phi d\vec{\xi} + \oint_{\partial E} (\mathcal{G}_n^* - \mathcal{H}_n^*) \phi ds - \int_E \vec{F}(U, \vec{S}) \cdot \nabla_\xi \phi d\vec{\xi} &= 0.\end{aligned}\tag{14}$$

Where $\vec{\mathfrak{U}}_n^*$ is the numerical flux for the additional equation and $\mathcal{H}_n^* = \mathcal{H}_n^*(U^+, U^-, \vec{S}^+, \vec{S}^-)$ stands for the numerical flux function for the viscous terms. As introduced in [2], we choose

$$\mathfrak{U}_n^* = (\alpha_{\text{visc}} U^+ + (1 - \alpha_{\text{visc}}) U^-) \vec{n},\tag{15}$$

$$\mathcal{H}_n^* = \left(\alpha_{\text{visc}} \mathcal{H}_n(U^+, \vec{S}^+) + (1 - \alpha_{\text{visc}}) \mathcal{H}_n(U^-, \vec{S}^-) \right),\tag{16}$$

\vec{n} is the surface normal pointing outwards. Only vectorial components of the solution U are rotated (like velocity or impulse), scalars are not affected. α_{visc} is chosen to be $\frac{1}{2}$, this method is named BR1 (first method of Bassi and Rebay [2]). In this work also the temperature gradient ΔT is calculated with the BR1 method because it is needed for the heat flux (see subsection 2.1).

2.3 Equation of State

In this work CoolProp [4] is used to provide the highly realistic EOS. It uses the Helmholtz energy functions which are described shortly in this subsection. The EOS can be written as following:

$$\frac{a(T, \rho)}{RT} = \frac{a^0(T, \rho) + a^r(T, \rho)}{RT} = a^0(\tau, \delta) + a^r(\tau, \delta),\tag{17}$$

where τ is the inverse reduced temperature T_r/T and the reduced density ρ/ρ_r is denoted by δ . The reduced parameters are normally the critical states [4][19]. a^0 represents the ideal gas part and a^r the residual part. The EOS is written as a function depending on the density and temperature of a fluid. The ideal gas part can be written as

$$a^0(T, \rho) = -1 + \ln \frac{\rho T}{\rho_0 T_0} + \frac{h_0^0}{RT} - \frac{s_0^0}{R} + \frac{1}{T} \int_{T_0}^T \frac{c_p^0(T)}{R} dt - \int_{T_0}^T \frac{c_p^0(T)}{RT} dt.\tag{18}$$

ρ_0 , T_0 , h_0^0 and s_0^0 are reference state parameters. R is the specific gas constant and c_p^0 is function at constant pressure. The residual part contains two parts, a polynomial and an exponential,

$$a^r(\tau, \delta) = \sum_{i=1}^{I_{\text{Pol}}} n_i \tau^{t_i} \delta^{d_i} + \sum_{i=I_{\text{Pol}}+1}^{I_{\text{Pol}}+I_{\text{Exp}}} n_i \tau^{t_i} \delta^{d_i} \exp(-\delta^{p_i}).\tag{19}$$

n_i , t_i , d_i and p_i are coefficients which are determined by nonlinear fits of analytically or experimentally gained thermodynamic properties. I_{Pol} are the number of polynomial terms and I_{Exp} of the exponential ones, depending on the thermodynamic region and the fluid. Using eq. (17) all other state variables, like pressure, speed of sound, etc., can be evaluated by analytic differentiation with respect to the density and temperature.

For the Navier-Stokes equation density and internal energy e can be calculated from the conservative variables: $e = E - 0.5 \vec{v}^2$. Since the Helmholtz energy formulation uses the density and temperature as input the correlation between these quantities must be evaluated. An easy way to do this is to iterate the

temperature to find the corresponding internal energy. The starting point is the given internal energy e_g and the density ρ_g :

$$f(T) = e_g - e(\rho_g, T) = 0. \quad (20)$$

The bisection method is used in this work to find the correct temperature. After the bisection algorithm, the corresponding temperature is known and with ρ_g and T all other parameters can be calculated easily. This procedure is very robust but numerically very time consuming. In section 3 an idea is explained, which uses this method in a preprocessing step to build a high accurate table for the EOS, which can be later used during the calculation and postprocessing.

2.4 Riemann Solvers

As seen in eq. (12), $\mathcal{G}_n^* := \mathcal{G}_n^*(U^+, U^-)$ needs to be evaluated. \mathcal{G}^* is calculated with a Riemann solver. On the cell surface, there are two distributions to the solution, U^+ and U^- , from the two conjunct cells connected by the surface. The intermediate state is calculated by a Riemann solver. Basic and detailed information for Riemann solvers can be found in [20] and the matrices needed for the Roe-Riemann solver for realistic EOS can be found in [5]. The states U^+ and U^- are rotated to the surface coordinate system. The corresponding numerical flux is back rotated to the physical space.

2.4.1 Local Lax Friedrich (LF)

This is one of the simplest Riemann solvers and approximates the rarefaction wave fan as a single rarefaction wave with the maximum signal speed and neglects the contact discontinuity. The shock wave is resolved also with the maximum signal speed. An estimate for the fasted wave speed is guessed as the following

$$s = \text{MAX}(|u^-|, |u^+|) + \text{MAX}(a^-, a^+). \quad (21)$$

With this maximum wave speed the numerical flux for the local Lax-Friedrich Riemann solver can be computed

$$\mathcal{G}_n^* = 0.5 \cdot (F_n^{a-} + F_n^{a+}) - s \cdot (U_n^+ - U_n^-). \quad (22)$$

2.4.2 Harten-Lax-van Leer-Contact (HLLC)

This Riemann solver can additionally resolve the contact discontinuity. For the wave speed two estimates are needed

$$s^- = \text{MIN}(u^-, u^- - a^-, u^- + a^-, u^+, u^+ - a^+, u^+ + a^+), \quad (23)$$

$$s^+ = \text{MAX}(u^-, u^- - a^-, u^- + a^-, u^+, u^+ - a^+, u^+ + a^+). \quad (24)$$

Also an estimation to the wave speed for the contact discontinuity is needed:

$$s_* = \frac{p^+ - p^- + \rho^- u^- (s^- - u^-) - \rho^+ u^+ (s^+ - u^+)}{\rho^- (s^- - u^-) - \rho^+ (s^+ - u^+)}. \quad (25)$$

With this the corresponding numerical flux can be calculated

$$\mathcal{G}_n^* = \begin{cases} F^{a-} & \text{if } 0 \leq s^- \\ F_*^{a-} & \text{if } s^- \leq 0 \leq s_* \\ F_*^{a+} & \text{if } s_* \leq 0 \leq s^+ \\ F^{a+} & \text{if } 0 \geq s^+ \end{cases} \quad (26)$$

where the fluxes in the star regions are given as

$$F_*^{a\pm} = F^{a\pm} + s^\pm (U_*^\pm - U^\pm). \quad (27)$$

The solutions in the star regions can be calculated with

$$U_*^\pm = \rho^\pm \begin{pmatrix} \frac{s^\pm - u^\pm}{s^\pm - s_*} \\ 1 \\ s_* \\ v^\pm \\ w^\pm \\ \frac{E^\pm}{\rho^\pm} + (s_* - u^\pm) \left[s_* + \frac{p^\pm}{\rho^\pm(s^\pm - u^\pm)} \right] \end{pmatrix}. \quad (28)$$

2.4.3 Roe

The Roe-Riemann solver is well known and is computing the numerical flux as follows

$$\mathcal{G}_n^* = \frac{1}{2}(F^- + F^+) - \frac{1}{2}|\Lambda|R_*L_*(U^+ - U^-), \quad (29)$$

where Λ are the eigenvalues of the system

$$\Lambda = \begin{pmatrix} u - a \\ u \\ v \\ w \\ u + a \end{pmatrix}. \quad (30)$$

R_* are the right eigenvectors

$$R_*(\bar{\mathbf{U}}) = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ (u - a) & u & 0 & 0 & (u + a) \\ v & v & 1 & 0 & v \\ w & w & 0 & 1 & v \\ (H - ua) & H - \rho a^2 e_p & v & w & (H + ua) \end{pmatrix}, \quad (31)$$

and L_* are the left eigenvectors

$$L_*(\bar{\mathbf{U}}) = \begin{pmatrix} \frac{u+a}{2a} - \frac{H-\mathbf{u}^2}{\rho a^2 e_p} & \frac{-1}{2a} - \frac{u}{\rho a^2 e_p} & \frac{-v}{\rho a^2 e_p} & \frac{-w}{\rho a^2 e_p} & \frac{1}{\rho a^2 e_p} \\ \frac{H-\mathbf{u}^2}{\rho a^2 e_p} & \frac{u}{\rho a^2 e_p} & \frac{v}{\rho a^2 e_p} & \frac{w}{\rho a^2 e_p} & \frac{-1}{\rho a^2 e_p} \\ -v & 0 & 1 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ -\frac{u-a}{2a} - \frac{H-\mathbf{u}^2}{\rho a^2 e_p} & \frac{1}{2a} - \frac{u}{\rho a^2 e_p} & \frac{-v}{\rho a^2 e_p} & \frac{-w}{\rho a^2 e_p} & \frac{1}{\rho a^2 e_p} \end{pmatrix}. \quad (32)$$

To calculate $e_p = \frac{\partial e(\rho, p)}{\partial p}$ a finite difference approach is used in this work. The matrices for R_* and L_* can be found in [5].

2.5 FV-Subcell Approach

As stated earlier a shock capturing technique is needed which can handle the high gradients during phase change and ordinary shocks. In this work a finite volume (FV) sub-cell approach is used. The Persson indicator [14] is used to switch between the two methods in a cell during each step. The number of DOF are the same for both methods as seen in figure 1. A conservative projection is used to transform the solution between DG to FV cell representation (and vice versa) if necessary. The derivation of the FV approach starts with

$$\int_{e_{ijk}} JU_t d\xi + \int_{e_{ijk}} \nabla_\xi \cdot \vec{\mathcal{F}}(U, \nabla_\xi U) d\xi = 0 \quad i, j, k = 0, \dots, N. \quad (33)$$

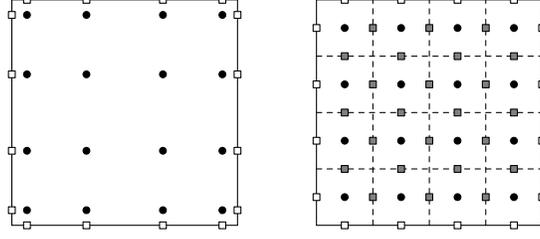


Figure 1: DOF distribution for a DG (left) and FV cell



Figure 2: Inner-cell and over-interface reconstruction

The reference element is split into $(N + 1)^3$ equidistant FV cells where e_{ijk} donates one cell. Using again the integration by parts rule the FV method is written as

$$\int_{e_{ijk}} JU_t d\xi + \int_{\partial e_{ijk}} g_n^*(U, \nabla_\xi U) dS_\xi = 0$$

where the nodal value U_{ijk} are constant mean values of ijk -th FV sub-cell. g_n^* is the numerical flux for each FV element. The FV approach can be written in a discrete way as

$$\begin{aligned} J_{ijk}(U_{ijk})_t &= \frac{N+1}{2} \left[-g_{i-\frac{1}{2},j,k}^{*,\xi^1} - g_{i+\frac{1}{2},j,k}^{*,\xi^1} \right] \\ &+ \frac{N+1}{2} \left[-g_{i,j-\frac{1}{2},k}^{*,\xi^2} - g_{i,j+\frac{1}{2},k}^{*,\xi^2} \right] \\ &+ \frac{N+1}{2} \left[-g_{i,j,k-\frac{1}{2}}^{*,\xi^3} - g_{i,j,k+\frac{1}{2}}^{*,\xi^3} \right]. \end{aligned}$$

The subscript n for the g_n^* was omitted here for the sake of simplicity. A second-order reconstruction is used in this work, see figure 2. Also if a DG and FV cell are adjacent to each other the reconstruction is done by projection of the face solution of the DG face to a FV distributed solution. The gradients for the reconstruction are limited with the MinMod-Limiter which gives this approach the total variation diminishing (TVD) property. For the viscous fluxes in the FV method the middle value of two neighboring gradients are used. The gradients are constructed and limited in primitive variables $U_p = (\rho, u, v, w, T)^T$ where T is the temperature.

2.6 Time Integration

For time integration a low storage explicit Runge-Kutta 4th order (LSERK4) algorithm is used. In algorithm 1 K has the same dimension as the solution vector U . The constants a_i and b_i can be taken from table 1.

```

K = 0;
for i ← 1 to 5 do
    K = aiK + Δt Ut(U);
    U = U + biK;
end

```

Algorithm 1: low storage Runge-Kutta 4th order algorithm

Table 1: Coefficients for the LSERK4 algorithm

i	a_i	b_i
1	0	$\frac{1432997174477}{9575080441755}$
2	$-\frac{567301805773}{1357537059087}$	$\frac{5161836677717}{13612068292357}$
3	$-\frac{2404267990393}{2016746695238}$	$\frac{1720146321549}{2090206949498}$
4	$-\frac{3550918686646}{2091501179385}$	$\frac{3134564353537}{4481467310338}$
5	$-\frac{1275806237668}{842570457699}$	$\frac{2277821191437}{14882151754819}$

The time step Δt is calculated with the advection time step

$$\Delta t_{\min}^{\text{adv}} \sim \frac{1}{\lambda_{\max}^{\text{adv}}} \frac{\Delta x}{2N + 1}, \quad (34)$$

where Δx is the element length and $\lambda_{\max}^{\text{adv}}$ the maximum eigenvalue of the flux Jacobian matrix for advection. Also the diffusion time step is needed

$$\Delta t_{\min}^{\text{d}} \sim \frac{1}{\lambda_{\max}^{\text{d}}} \left(\frac{\Delta x}{2N + 1} \right)^2, \quad (35)$$

where $\lambda_{\max}^{\text{d}}$ is the maximum eigenvalue of the diffusion matrix [7]. Δt is the minimum out of these two time steps.

2.7 Parallelization

The parallelization of the solver is done with MPI routines. It is designed to handle one DG element per core at its limit. A communication-hiding technique is used and is explained in more detail in [1, 13]. For this kind of FV shock capturing the memory layout did not change and the parallelization technique can be easily adapted. The reconstruction for the 2nd order FV scheme is also done over MPI interfaces.

3 Efficient implementation of the Equation of State

To resolve effects like cavitation and condensation in a dense gas approach, a realistic EOS has to close the Navier-Stokes equations. In this work the CoolProp 4.2.6 library [4] is used. For water, as example, the IAPWS-IF95 [21] standard with thermodynamic equilibrium in the two-phase region is used, but with CoolProp over 100 fluids can be evaluated. The easiest way to implement the EOS would be to directly couple CoolProp with the DG solver during calculation. This involves also the temperature iteration explained in subsection 2.3, which is not part of CoolProp. The library can handle density and internal energy as an input, but in this work control of the iteration is needed.

Table 2: Comparison of evaluation time (μs) perfect gas and realistic EOS

	Vapor	Two-Phase	Liquid
Perfect Gas	0.01712	0.01712	0.01712
Realistic	232.015	2392.149	260.016
Table	0.32402	0.36802	0.30002

If the evaluation time between a perfect gas and realistic EOS is compared it is easy to see why a different approach is needed. Table 2 shows the evaluation time for density and internal energy as input parameters. Both types of EOS calculate, with these two inputs, the following output parameters: temperature, pressure, speed of sound, viscosity and heat conductivity. The evaluation of the perfect gas EOS is several orders of magnitude faster which means, by using the realistic EOS directly, the performance of the code will drop by this factor. This is not usable in a CFD context. In the next subsections an approach is explained, which reduces this factor extremely.

3.1 Table Approach

In this work a table approach is used to reduce the overhead of the realistic EOS evaluation. To build such a table, which stores the information of the realistic EOS, the minimum and maximum values for the input parameters define a two dimensional space. This area is divided into a $2^l \times 2^l$ equidistant elements. l stands for the level but normally the table build process starts at level 1. The boundaries for the table are defining the computational domain $\mathcal{T} = [\rho_{\min}, \rho_{\max}] \times [e_{\min}, e_{\max}]$. Each element has the size $t = [\Delta^l \rho, \Delta^l e] = [\frac{\rho_{\max} - \rho_{\min}}{2^l}, \frac{e_{\max} - e_{\min}}{2^l}]$ depending on the level. Also every element is mapped to the unit element $\mathcal{E} = [0, 1] \times [0, 1]$ with the coordinates ξ and η . For each cell a polynomial representation of the EOS in the unit element is build

$$q_h^p(\xi, \eta) = \sum_{i,j=0}^p \hat{q}_{ij} \psi_{ij}(\xi, \eta), \quad \hat{q}_{ij} = q(\xi_i, \eta_j). \quad (36)$$

The root of each element is the corner with both minimum values. \hat{q}_{ij} is evaluated with the help of the EOS library for the input values $\rho_{\text{root}} + \xi_i \Delta^l \rho$ and $e_{\text{root}} + \eta_j \Delta^l e$. The used nodal basis functions $\psi_{ij}(\xi, \eta)$ are build with Chebyshev-Gauss nodes. The polynomial degree p can vary for each element. In this work $p_{\max} = 10$ and the degree used by the element is the minimum polynomial degree which satisfies

$$\mathbb{L}_{\infty} = \max \left(\frac{q_h(\xi, \eta) - q(\xi, \eta)}{q(\xi, \eta)} \right)_{\infty} < \epsilon_t. \quad (37)$$

The maximum norm error-criterion is checked on 20×20 equidistant points in each element. In case the \mathbb{L}_{∞} -error of the polynomial approximation is not satisfied by an element this can be refined by dividing it in 2×2 equidistant elements. This is repeated until the error is lower or the maximum level is reached. $L = 32$ is the maximum level because the element localization, e.g., finding the root of an element, is done with a 64-bit number. Figure 3 shows the 1st and 2nd level of a table with the corresponding identification binary number N . This number is read from right to left. The first two digits correspond to level one. The second pair of numbers belongs to the second level and so on. Algorithm 2 shows a way to calculate the root values for each element by using the identification number N . Here l_e is the level of the element.

```

\rho_{\text{root}} = \rho_{\text{min}};
\epsilon_{\text{root}} = \epsilon_{\text{min}};
for  $l \leftarrow 1$  to  $l_e$  do
  |  $\rho_{\text{root}} = \rho_{\text{root}} + N[2^l - 1]\Delta^l \rho;$ 
  |  $\epsilon_{\text{root}} = \epsilon_{\text{root}} + N[2^l]\Delta^l \epsilon;$ 
end

```

Algorithm 2: Finding the root for a identification number N

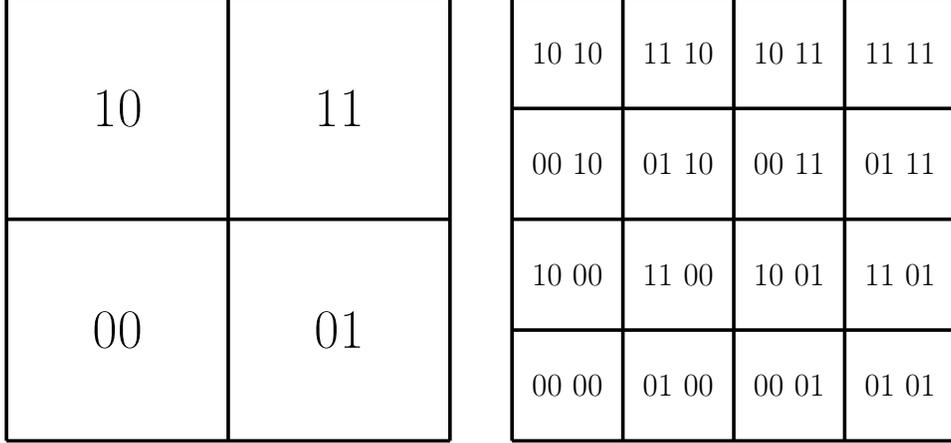


Figure 3: Level 1 and 2 with identification number

$N[x]$ denotes the bit at the x position. The return value is 0 or 1. A schematic of the quadtree is shown in figure 4 with the connections to the previous levels. Figure 5 shows a table for water with $\rho_{\min} = 1 \text{ kg/m}^3$ and $\rho_{\max} = 1330 \text{ kg/m}^3$ defining the x -axis and $e_{\min} = -9900 \text{ J}$ and $e_{\max} = 4.056 \cdot 10^6 \text{ J}$ is the range of the y -axis. The building error was set to $\epsilon_t = 1 \cdot 10^{-7}$ and $L = 17$ for this table. The quantity q for the error calculation equals the temperature.

Because of the higher gradients at the liquid and vapor saturation line the algorithm has to refine there to reach the desired error. Since this can lead to a very large number of elements, a cut-cell approach is also implemented in the table algorithm. To find a cut-cell the saturation temperature is needed. This can be found with a given density and the vapor quality χ for the liquid or vapor saturation line, where $\chi = 0$ and $\chi = 1$, respectively. The density is given by the minimum and maximum values for one element. Two points ($p_1 = (\rho_{\min}, \chi)$ and $p_2 = (\rho_{\max}, \chi)$) are found in the table plane for each saturation line which are connected by a linear line. This line is checked if it is cutting the element. Of course this is a linear approximation to a curved line, but at higher levels the saturation line gets better approximated because the elements are becoming small. In figure 6 the four possible cut-cell types are illustrated neglecting the rotation. By cutting the element triangles are created. Inside this triangle a polynomial representation of the EOS is build with the same degree as the element. Also for the triangular cut-cells the maximum error is evaluated and if it holds condition 37 no more refinement is needed. If both kind of cells reach the needed criterion the polynomial representation of the quadric cell is used.

To evaluate the table during computation, a fast quadtree integer-based bisection described in [12] is used to find the corresponding element. In the element the polynomial basis (see eq. (36)) is evaluated to find the needed quantity (e.g. temperature). The time to calculate the needed quantities temperature, pressure, speed of sound, viscosity and heat conductivity from density and internal energy for the table approach can be seen in table 2 compared to the perfect gas equation and the EOS library.

3.2 Parallelization

The building process of the table is fully parallelized. It can be build with any number of processors. The root core is setting up the level by creating the unique identification number per element needed for the level. The amount of numbers is divided by the number of cores. Each core knows the minimum and maximum of the input variables (e.g. density and internal energy). The identification numbers and current level are broadcast by the root core to every core. With these numbers each core can evaluate the polynomial functions to approximate the EOS for each element it received from the root core. When all elements are done the cores send the data back to the root. They send back following information: maximum error norm for each element, polynomial representation for quadric and cut elements and if a refinement is necessary. The root collects all this information and builds each level. If a new refinement level is needed it builds the identification numbers for the new level and sends it evenly distributed to the used cores. If the number of

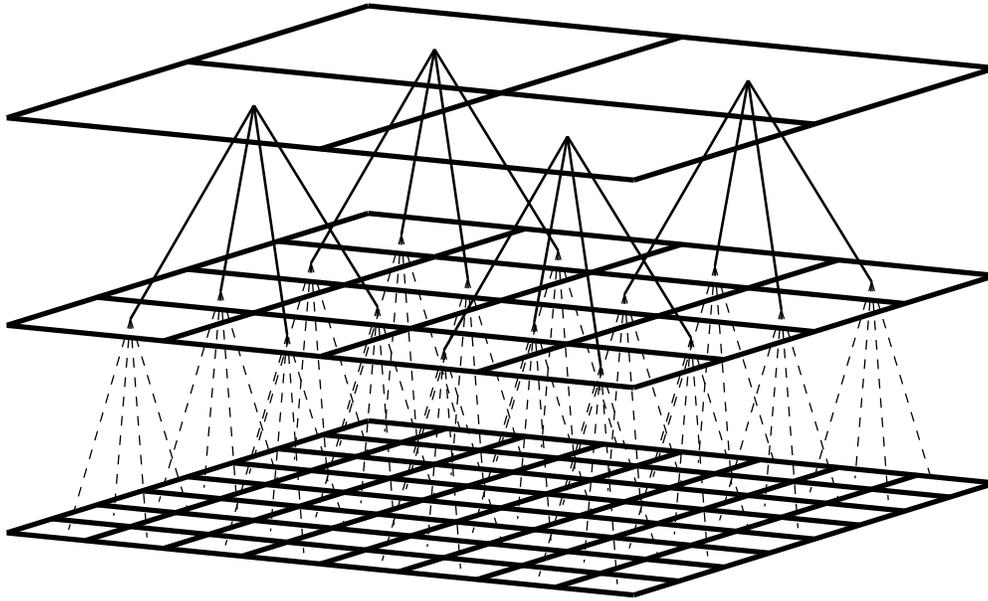


Figure 4: Schematic quadtree with level connections

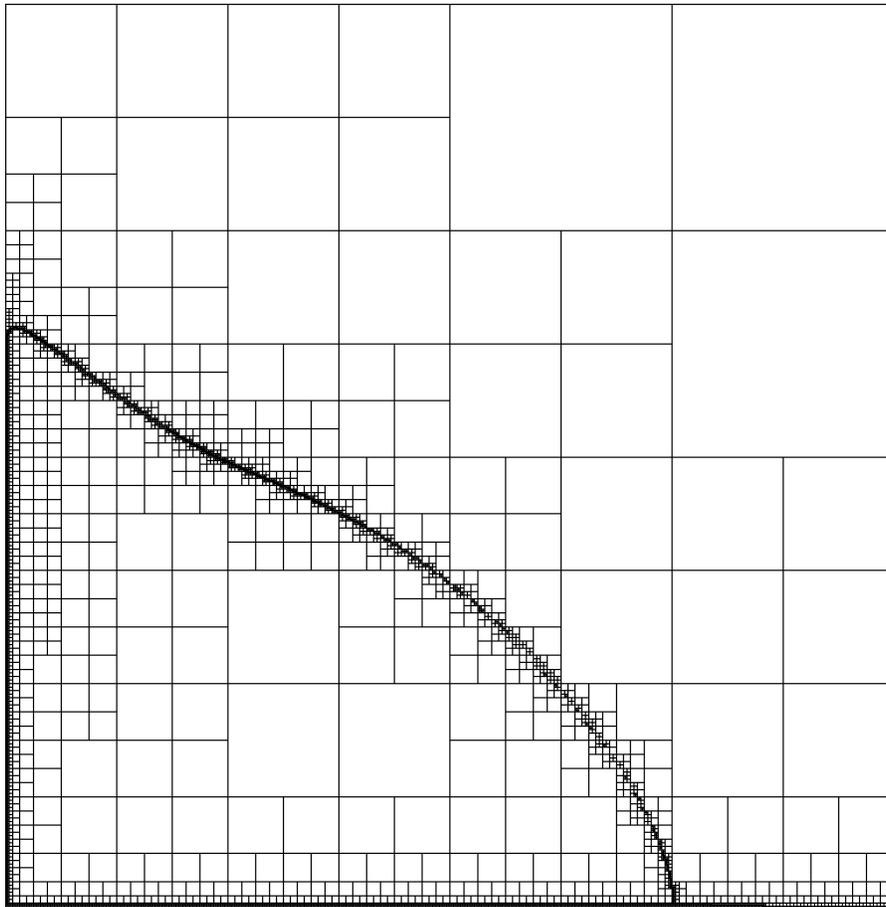


Figure 5: A table for water. x -axis(ρ)= $[1, 1330]$ kg/m^3 and y -axis(e)= $[-9900, 4.056 \cdot 10^6]$ J

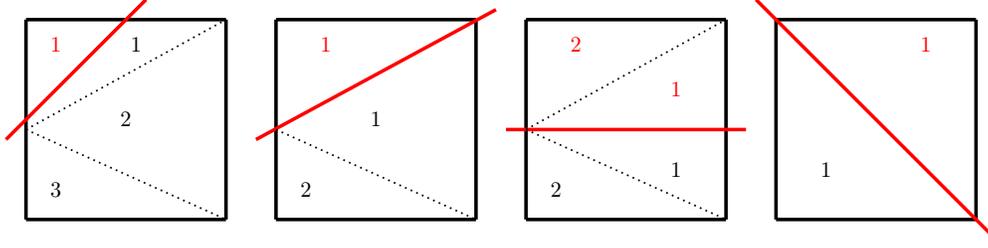


Figure 6: 4 possible cut-cells types

cores is higher than the number of elements some of the cores idle.

3.3 Table types

During calculation different table types are needed. The tables used during calculation in this work are listed in table 3. The table shows the input variables and the output quantities. The used table dimensions are stated in x and y direction. The used target error for the building process and the used numbers of cores are also listed in the table. With these used parameters the corresponding building time and the area size in percentage which could not be resolved can be found in the table information. Even with the highest level some elements still do not satisfy the error criterion. Since the elements are getting smaller each level the uncovered area is very small. In the 'purpose'-row C stands for the conservative variables and P for the primitive ones. The arrow shows the conversion directions for which the table is used. One table is used for both directions. For the $C \rightarrow P$ calculation first the ' $(\rho, e) \rightarrow T$ '-table is evaluated and with the received

Input Output	Table 3: Tables for water used in this work			
	(ρ, e) T	$(\frac{1}{\rho}, e)$ T	(T, ρ) e, p, a, μ, λ	(T, p) ρ, e
x -value range	[1, 1330] kg/m ³	[1, 10000] m ³ /kg	[276, 1273] K	[276, 1273] K
y -value range	[-9.9, 4056] kJ	[-9.9, 4056] kJ	[0.0001, 1330] kg/m ³	[0.001, 10000] MPa
error ϵ_t	1×10^{-7}	1×10^{-7}	1×10^{-7}	1×10^{-6}
build time [min]	14	15	5	10
# cores	4800	4800	4800	1200
area not covered [%]	1.5×10^{-2}	3.0×10^{-3}	9.7×10^{-5}	1.2×10^{-5}
Purpose	$C \rightarrow P$	$C \rightarrow P$	$C \leftrightarrow P$	BC
size [GB]	0.32	1.4	3.9	2.2
L	17	19	22	23
Valid elements [%]	44	51	39	38

temperature the ' $(T, \rho) \rightarrow (e, p, a, \mu, \lambda)$ '-table calculates the primitive quantities. In the other direction the density and temperature are the primitive inputs and the internal energy is an output. The purpose 'BC' means boundary condition and this table is used for the isothermal wall condition. The size specifies the needed space on a hard disk. Keep in mind that all the calculated elements are stored regardless of the error. The percentage of valid elements shows the amount of elements which satisfy the error criteria. This means that the needed amount of memory during calculation is lower. For post-processing a table can be used with a lower error and more quantities which are not needed during calculation. The lower error reduces the needed disk space drastically. See table 4 for more information.

4 Results

In this chapter the calculation results are shown. At first the high-order convergence for pure DG calculation and the one for mixed DG and FV calculations are shown. 1D and 3D convergence is presented. For validation 1D calculations are taken from [10, 17]. Some Riemann problems and 1D cavitation as well as condensation

Input	(T, ρ)
Output	$e, p, a, \mu, \lambda, s, q$
x-value range	$[0.0001, 1330] \text{ kg/m}^3$
y-value range	$[274, 1273] \text{ K}$
error	1×10^{-3}
build time [min]	10
# cores	48
area not covered [%]	0.19
Purpose	Post-processing
size [GB]	0.2
L	14
Valid Cells [%]	51

is shown. For the use-case simulation a NACA 0012 in water is calculated in 2D. For scaling tests a 3D channel flow with cavitation is shown. All variables values are given in SI units.

4.1 Convergence

For convergence investigation a manufactured solution is used in 1D and 3D. All convergence tests are done with periodic boundary condition and the perfect gas EOS for validation purpose. The advection convergence is achieved by setting the viscosity $\mu = 0$. For the diffusion a value for μ is chosen so that the viscous terms dominate.

4.1.1 1D

A sine wave is transported from left to right in the 1D simulation where the length of the mesh is 2 m. Three different setups are shown. The first is a pure DG mesh (table 5). The convergence for the high-order scheme is achieved and also the one for the 2nd order FV scheme as well as for the mixed method. Only the density is shown but all other variables converge, too.

Table 5: L^2 errors and convergence rates of the density for 1D advected sinus wave for pure DG sub-cells method with a polynomial degree of the DG elements ranging from $N = 3$ and $N = 5$.

poly. degree	cells	advection		diffusion		theor. order
		error	order	error	order	
N=3	24	8.17e-08		7.77e-08		4
	48	4.80e-09	4.09	5.23e-09	3.89	
	96	2.99e-10	4.01	2.89e-10	4.18	
	192	1.87e-11	4.00	1.81e-11	4.00	
	384	1.19e-12	3.98	1.47e-12	3.63	
N=5	24	1.15e-11		1.23e-11		6
	48	1.79e-13	6.01	1.75e-13	6.13	
	96	9.45e-14	****	1.18e-13	****	
	192	1.80e-13	****	8.42e-14	****	
	384	3.55e-13	****	2.24e-13	****	

To check the convergence for the FV scheme all elements are forced to be represented as FV sub-cells. As seen in table 6 a order of up to 1.69 is reached. The second order cannot be achieved because a MinMod limiter is used. This decrease of the theoretical order is well known for a reconstruction with TVD limiter.

Table 6: L^2 errors and convergence rates of the density for 1D advected sinus wave for pure FV sub-cells method with a polynomial degree of the DG elements ranging from $N = 3$ and $N = 5$.

poly. degree	cells	advection		diffusion		theor. order
		error	order	error	order	
N=3	24	1.65e-03		3.61e-05		2
	48	5.14e-04	1.68	1.20e-05	1.58	
	96	1.60e-04	1.69	3.99e-06	1.59	
	192	5.00e-05	1.67	1.31e-06	1.60	
	384	1.57e-05	1.67	4.30e-07	1.61	
N=5	24	8.37e-04		1.91e-05		2
	48	2.59e-04	1.69	6.31e-06	1.60	
	96	8.09e-05	1.68	2.08e-06	1.60	
	192	2.54e-05	1.67	6.84e-07	1.61	
	384	8.00e-06	1.67	2.23e-07	1.62	

Table 7: L^2 errors and convergence rates of the density for 1D advected sinus wave for mixed DG and FV sub-cells method with a polynomial degree of the DG elements ranging from $N = 3$ and $N = 5$.

poly. degree	cells	advection		diffusion		theor. order
		error	order	error	order	
N=3	24	1.26e-03		3.46e-05		2
	48	4.45e-04	1.50	1.04e-05	1.74	
	69	1.27e-04	1.81	2.88e-06	1.85	
	192	3.61e-05	1.81	7.33e-07	1.97	
	384	1.11e-05	1.71	1.85e-07	1.98	
N=5	24	7.43e-04		1.73e-05		2
	48	2.26e-04	1.72	4.94e-06	1.81	
	96	5.98e-05	1.92	1.30e-06	1.92	
	192	1.82e-05	1.72	3.29e-07	1.98	
	384	5.47e-06	1.73	8.40e-08	1.97	

For the third test in 1D a mixed mesh is used. This means that every other element is FV and the rest DG. By comparing the order of convergence for advection and diffusion we see the influence of the gradients for the viscous flux calculation (see subsection 2.5). The second order can be reached because of the combination of the DG method and central evaluated gradients. The advection part still reaches around 1.7 because of the TVD MinMod limiter, see table 7.

4.1.2 3D

For the 3D convergence test a diagonal moving sine wave in a cubic mesh with an edge length of 2 m is used. For the mixed element type a 3D check-board pattern is used. The reason why the convergence rate is not as good as in 1D is the lower resolution. With the 48^3 elements and a polynomial degree of $N = 3$ this makes a total of 7 Mio DOF. Doubling each direction would lead to 60 Mio DOF which would have been too costly to calculate. We can see that convergence rate converges to a reasonable value in table 8.

Table 8: L^2 errors and convergence rates of the density for 3D sinus wave for DG and FV sub-cells method with a polynomial degree $N = 3$ of the DG elements.

poly. degree	cells	advection		diffusion		theor. order
		error	order	error	order	
DG	12^3	2.28e-06		2.18e-06		4
	24^3	1.41e-07	4.02	1.43e-07	3.93	
	48^3	8.88e-09	3.99	9.90e-09	3.86	
FV	12^3	1.46e-03		2.50e-04		2
	24^3	5.20e-04	1.49	7.62e-05	1.71	
	48^3	1.59e-04	1.70	2.20e-05	1.79	
MIX	12^3	1.85e-03		2.06e-04		2
	24^3	6.99e-04	1.40	6.46e-05	1.67	
	48^3	2.38e-04	1.56	1.94e-05	1.74	

4.2 1D Validations

For further validation purpose a couple of 1D calculations are discussed in this section. In [10] five Riemann-Problems for water are described. All of them work with this implementation, but only RP-W1, RP-W3 and RP-W5 are shown. Two additional calculations are performed which produces cavitation by rarefaction and a condensation test-case taken from [17].

4.2.1 Riemann Problems

All Riemann-Problem calculations are performed on a 50 element DG mesh with a degree 3 polynomial. This leads to 200 DOF. The local Lax-Friedrichs Riemann solver is used in the paragraphs RP-W1 to RP-W5. The exact solution is indicated by the red color and taken from [10]. For all Riemann problems the FV sub-cell approach is active. The indicator is set to be very sensitive that some elements are calculated with the FV method.

RP-W1 The given states are $U_{\text{left}}^p = (985.9853, 6.4656, 2750736, 329.9096)$ and $U_{\text{right}}^p = (983.7899, 2.44908, 1091213, 332.8354)$ in primitive variables $U^p = (\rho, u, p, T)$. The initial location of the discontinuity is $x_i = 0.5$. Both states are in the liquid region. The calculation is performed to an end time of $t=2 \times 10^{-4} s$. In figure 7 the numerical and exact solution are compared. The agreement is very good for the used local Lax-Friedrich Riemann solver. But of course the intermediate contact is smeared by this kind of Riemann solver.

RP-W3 The given states are $U_{\text{left}}^p = (976.0968, -187.4091, 13279.97, 324.6175)$ and

$$U_{\text{right}}^p = (971.2215, -208.485, 40776.45, 349.4697)$$

again in primitive variables. The initial location of the discontinuity is $x_i = 0.5$. Both pressures are the vapor pressures for the given temperature. This means the initial states lie at the liquid-vapor zone but the intermediate states are in the liquid area. This test has high pressure jumps and also phase change occur. In figure 8 the numerical and exact solutions are compared. The agreement is very good. Some oscillations occur for the pressure. This is because of the use of a highly accurate real fluid EOS and is also seen in [10].

RP-W5 The given states are $U_{\text{left}}^p = (998.23739, 100.0, 100000.0, 293.0)$ and

$$U_{\text{right}}^p = (996.55634, 100.0, 100000.0, 300.0)$$

again in primitive variables. The initial location of the discontinuity is $x_i = -0.5$. It demonstrates a isolated moving contact discontinuity. As mentioned in [18] the use of highly non-linear EOS can produce oscillation in pressure and velocity for moving contact discontinuities. As seen in figure 9 this is also the case here. The

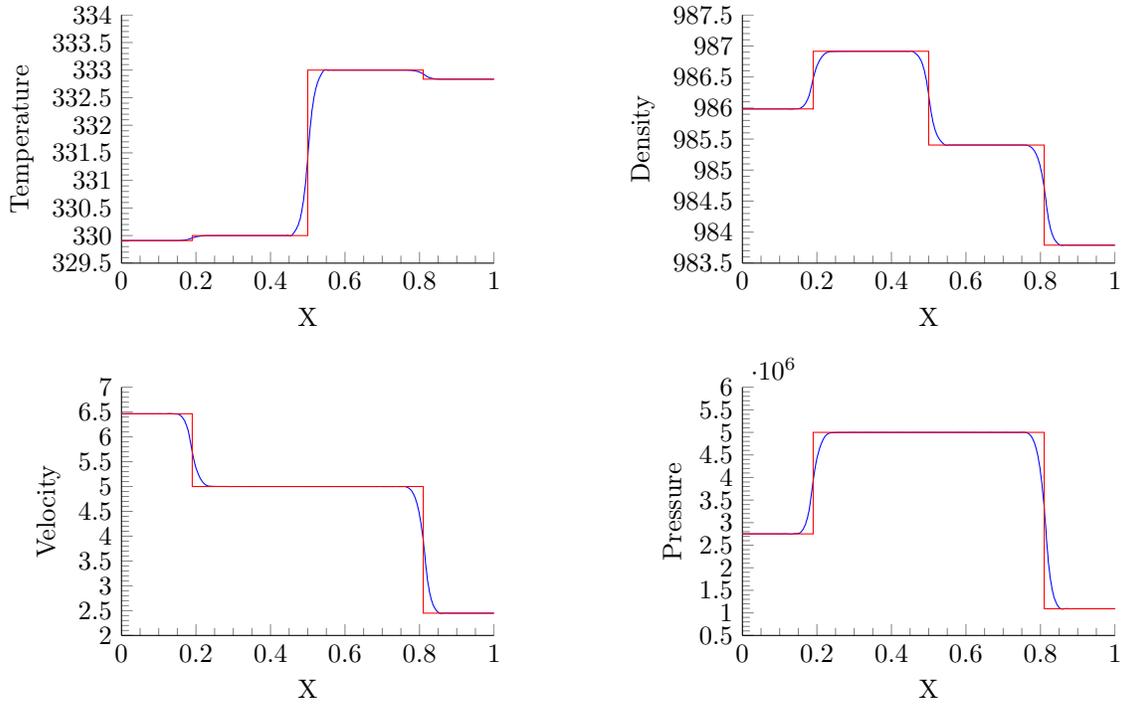


Figure 7: Riemann Problem RP-W1 at $t=2 \times 10^{-4}$ s: comparison of the exact solution (red) with the numerical solution (blue)

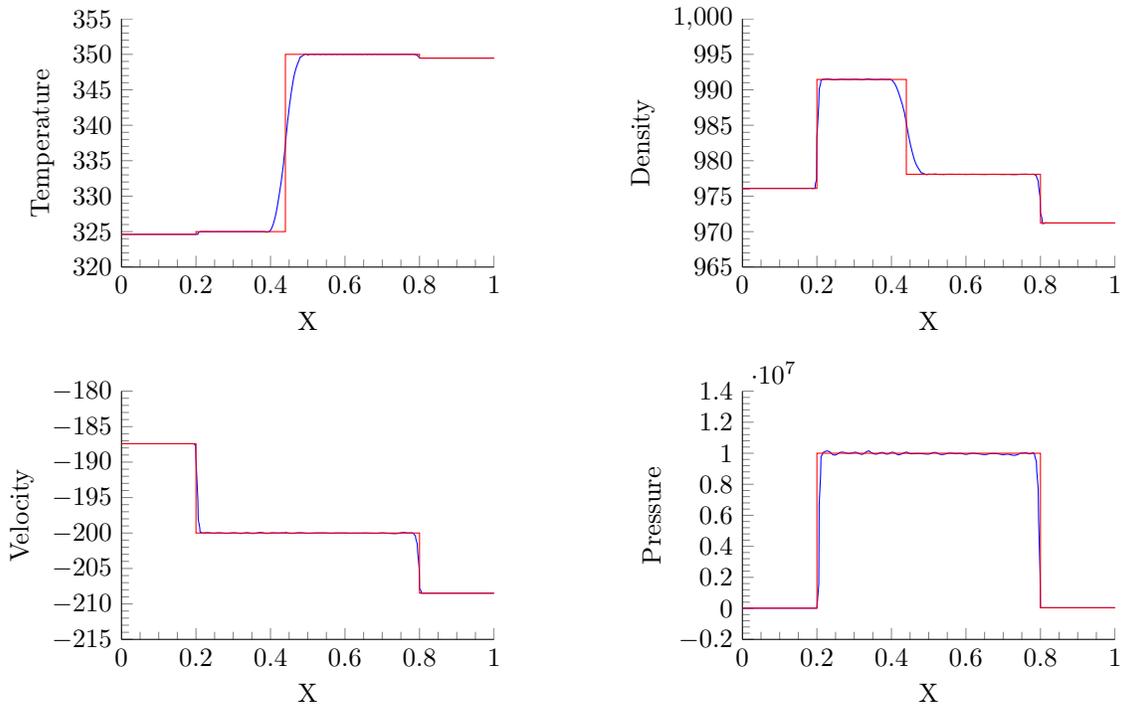


Figure 8: Riemann Problem RP-W3 at $t=3 \times 10^{-4}$ s: comparison of the exact solution (red) with the numerical solution (blue)

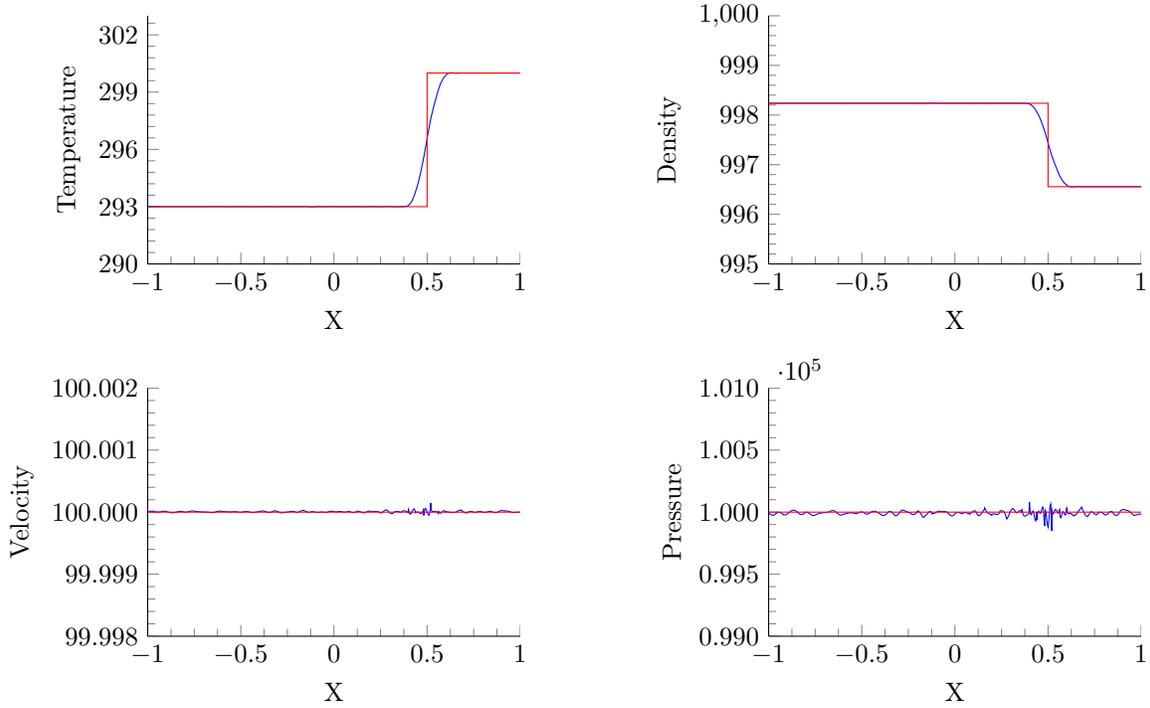


Figure 9: Riemann Problem RP-W5 at $t = 1 \times 10^{-2}s$: comparison of the exact solution (red) with the numerical solution (blue)

oscillations are more noticeable at the location of the discontinuity but almost not visible for the velocity. For the pressure they are around 1 per mill of the initialized data.

Different Riemann solvers If the LF, HLLC and Roe are compared we see the usual behavior. For the RP-W1 test case we see in figure 10 that the intermediate contact discontinuity is much better resolved with the HLLC and Roe Riemann solver. The Roe Riemann solver tends slightly more to oscillation but this vanishes when the resolution is increased by a factor of 4. This can be seen in figure 11. Also in this figure it is very well visible that all three Riemann solvers approximate the exact solution much better at a higher resolution. It has to be mentioned that for the RP-W3 the Roe Riemann solver is not producing stable results for a reasonable resolution.

4.2.2 Cavitation

The given states are $U_{\text{left}}^p = (997.0854, -10, 100000, 298)$ and $U_{\text{right}}^p = (997.0854, 10, 100000, 298)$ again in primitive variables. Here 2500 Elements and a polynomial degree of 3 is used. The initial position of the discontinuity is $x_i = 0$ for this test case. As seen in figure 12 due to the opposite velocity directions a cavitation area is produced. The sound speed drops by several orders of magnitude which is normal for multiphase problems and must be captured with the FV sub-cell approach. The results are in excellent agreement with the ones presented in [10].

4.2.3 Condensation

The given states are $U_{\text{left}}^p = (1200, 0, 766266513, 300)$ and $U_{\text{right}}^p = (500, 0, 3536.81, 300)$ again in primitive variables. The left state is filled with liquid and the right with liquid-vapor mixture with a vapor volume fraction of 0.5. The test case is taken from [17] but with a slightly adjustment to x_i because of the different sound speed velocities due to different EOS used in both methods. In this case $x_i = 0.64$. With this a very good result is reached compared to the original one. Figure 13 shows the obtained result.

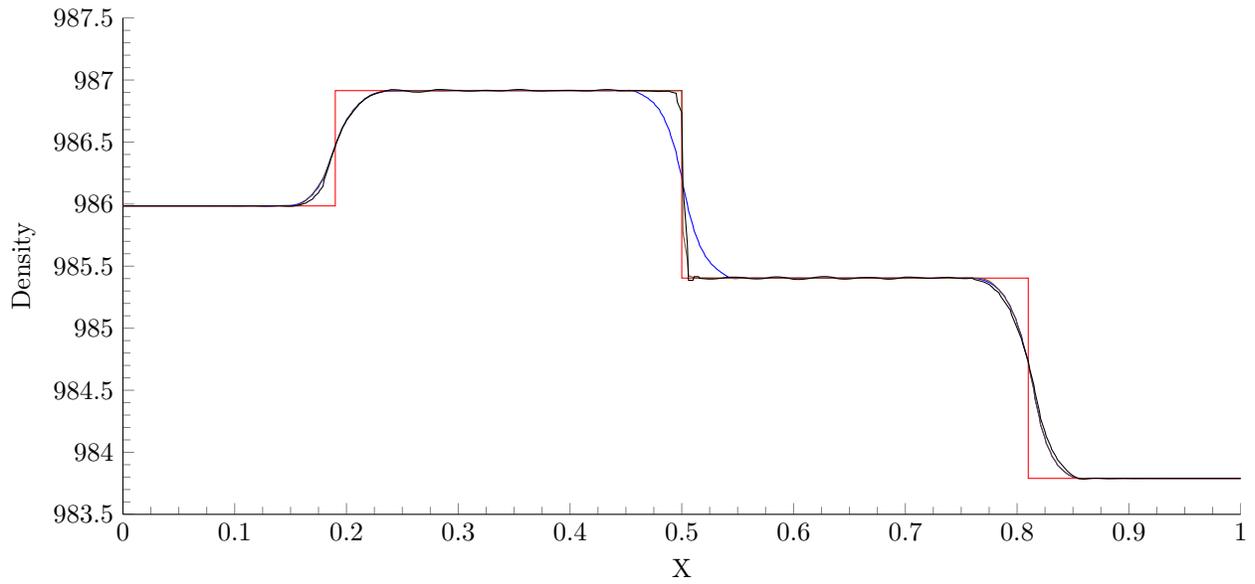


Figure 10: RP-W1 with LF (blue), HLLC (brown) and Roe (black) Riemann solvers. Exact solution is colored red. Here 50 elements were used with $N = 3$.

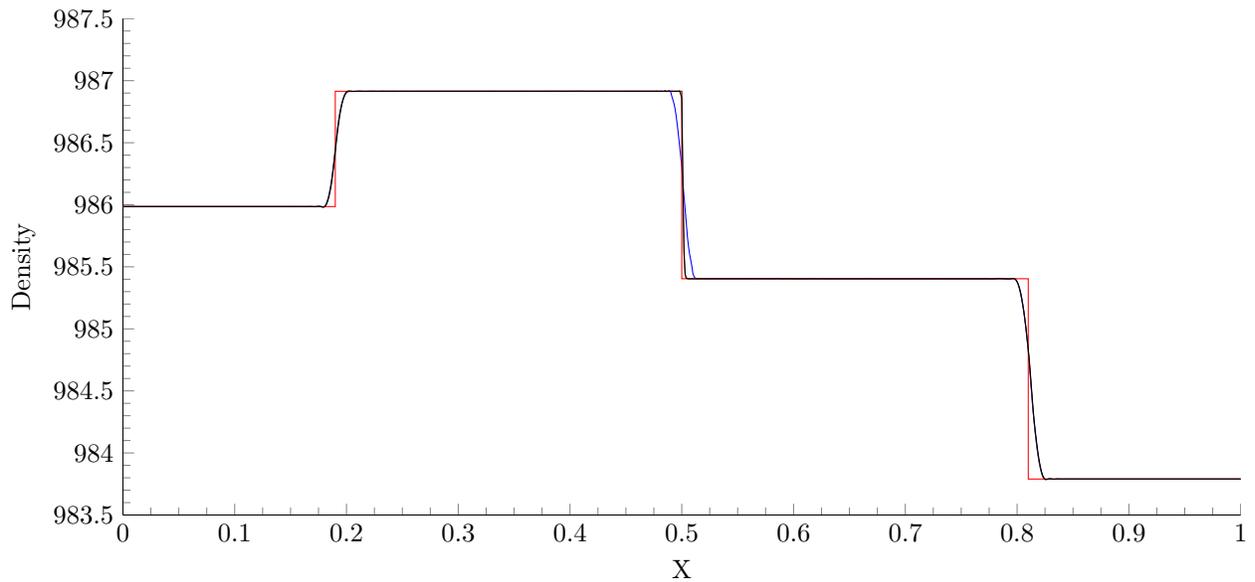


Figure 11: RP-W1 with LF (blue), HLLC (brown) and Roe (black) Riemann solvers. Exact solution is colored red. Here 200 elements were used with $N = 3$.

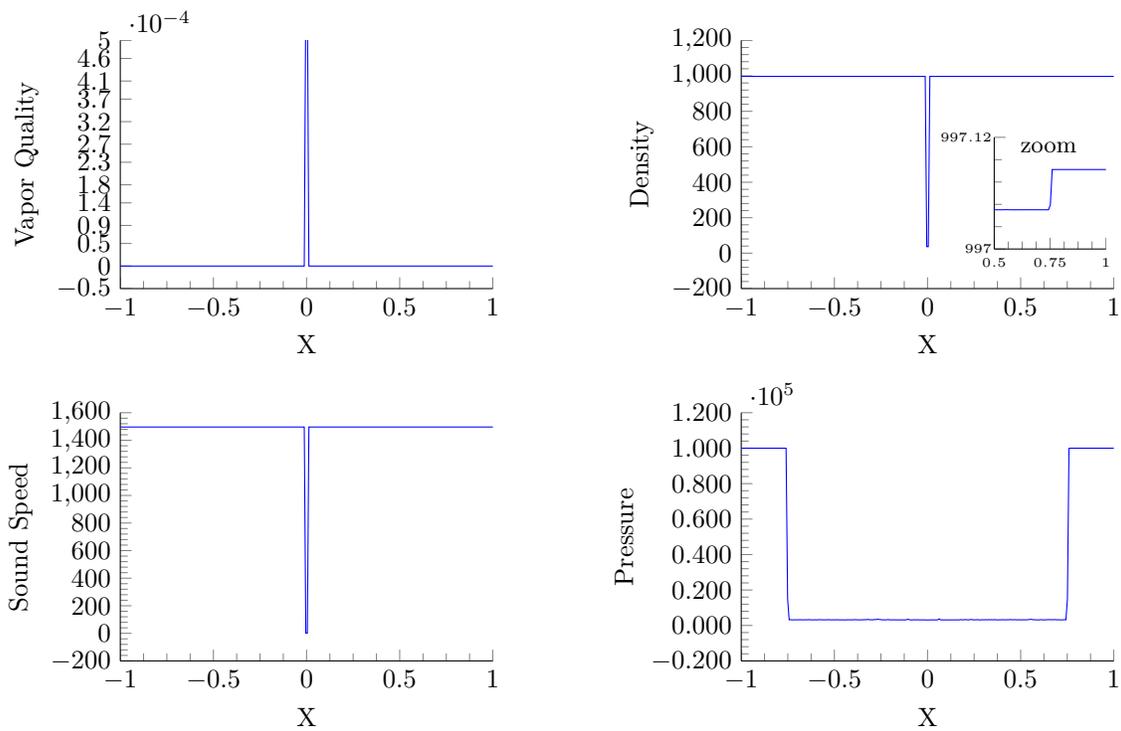


Figure 12: 1D Cavitation at $t=5 \times 10^{-4}s$

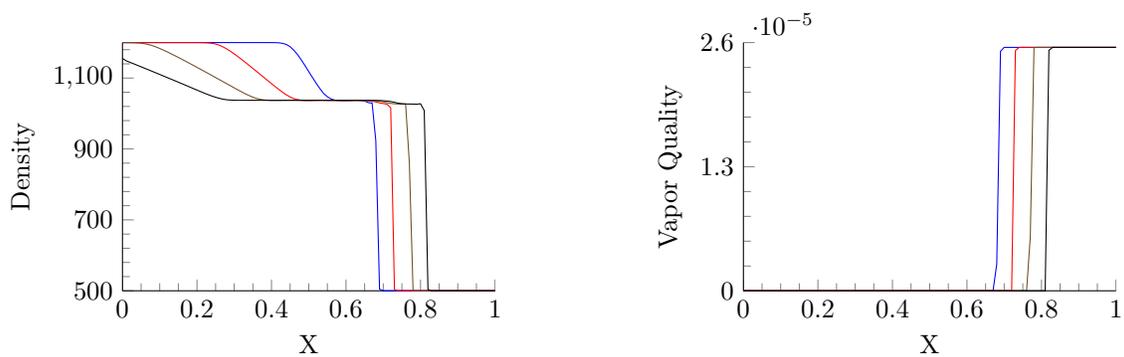


Figure 13: Condensation over time. 75 (blue), 150 (red), 225 (brown) and 300 μs (black)

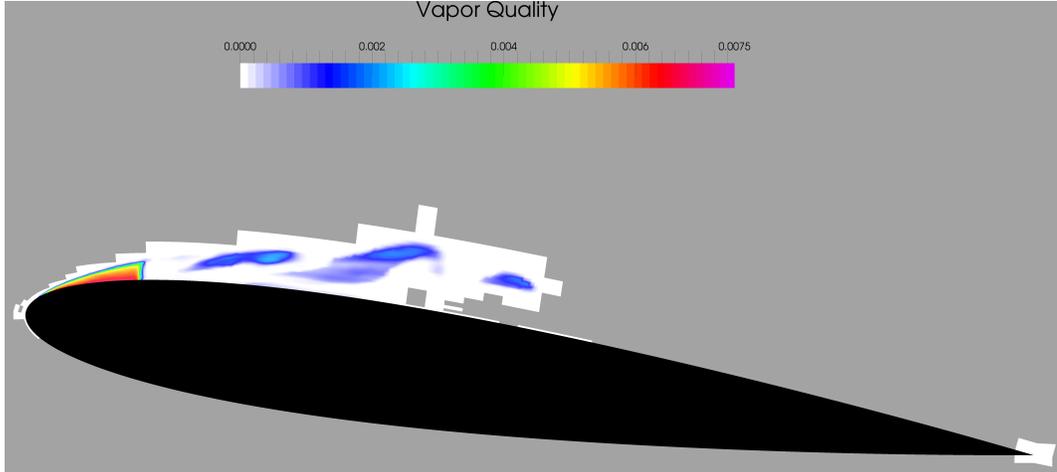


Figure 14: Hydro-Foil Naca-0012 with $T = 300$ K, $p = 5$ bar, $\sigma = 1$, $\alpha = 8^\circ$ at $t = 0.03423$ s. The white area highlights where the FV method is active.

4.3 2D Calculations: Hydro-Foil

For a 2D test case a Naca-0012 is used as a hydro-foil. It is surrounded by water at a temperature of 300 K and a pressure of 5 bar. The angle of attack is $\alpha = 8^\circ$ and the cavitation number $\sigma = 1$. A grid with 12194 elements is used with a polynomial degree of 4. This leads to ca. 1.5 Mio DOF. The simulation runs on 240 Cores. The chord length is 0.1334 m and the domain has a radius of one meter. This simulation is a show case to demonstrate the capability of the code to run such simulations with DG and FV coupling. Figure 14 shows the cavitation region which develops from the tip of the foil. The white area shows the elements which are using the FV method. all the rest of the computational domain uses the DG-SEM at this particular time frame. Since the Persson indicator is used to switch between these methods it changes for every timestep. But this calculation shows that the setup is well suited for resolving cavitation flows. In the near future the calculations will be compared to experimental data to verify the implementation.

4.3.1 3D Calculation: Scaling

To show the scaling of the code a 3D calculation is performed. A throttle is filled with liquid water at 100 bar and 330 K and the inlet pressure is higher than the outlet pressure. Due to this pressure gradient the water streams through the throttle and starts to cavitate. The inlet is pressurized with 300 bar and a temperature of 330 K. The outlet has a pressure of 100 bar and the same temperature as the inlet. The mesh for this calculation can be seen in figure 15. The mesh has a total of 46592 elements and the polynomial degree is set to three. For the scaling test a operating point is used where it cavitates in the throttle, which can be seen in figure 16. This was done to assure that the tabled EOS is evaluated in the liquid-vapor region because there is the highest refinement level. Also the coupling of the DG and FV scheme is active. Keep in mind, that this calculation is only for scaling. All boundary conditions, except the inlet and outlet, are walls. The scaling test is done on the CRAY XC40 (Hazel Hen) of the High Performance Computing Center Stuttgart (HLRS). The amount of cores ranges from 24 up to 6144. Each test the number of cores are doubled. As comparison of the needed simulation time the 24 core run is used as base to calculate the speed up. The scaling can be seen in figure 17. The minimum load per core are 7 elements which leads to 448 DOF per core at a core number of 6144. The lowest scaling compared to the 24 core run is still at 70 %. The decrease of the ideal speed-up is due to a load-imbalance between the cores. There are several reasons for this: the number of elements can not be divided evenly by the number of cores. Also the FV sub-cells are more computation-time consuming than the DG elements by a factor of around 1.5. The table approach also adds an imbalance because the time to evaluate the quantities depends on the level and polynomial degree of the evaluated table element. Also an element which has a boundary condition face needs slightly more time than a element without it. For an even better scaling a load-balancing algorithm has to be implemented.

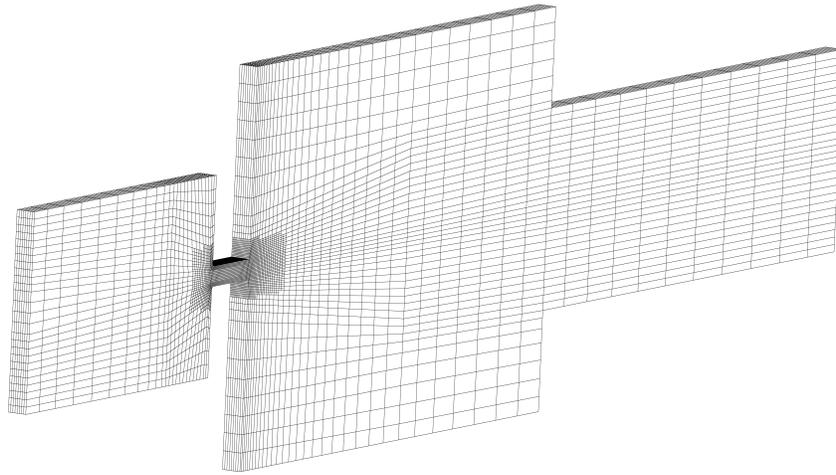


Figure 15: Mesh for the throttle flow with 46592 elements



Figure 16: Cavitation inside the throttle. Pitch black area shows the throttle wall. The gray-scale represents the vapor mass fraction for this slightly cavitating flow

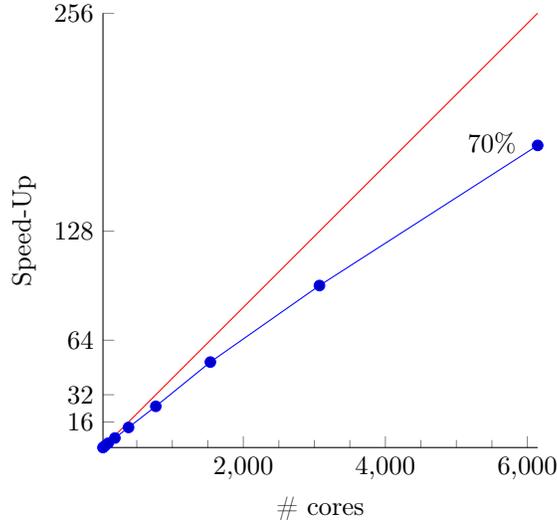


Figure 17: Scaling with $N = 3$ from 24 to 6144 cores

5 Conclusion and Future Work

We presented a DG-SEM solving the compressible Navier-Stokes equations with a highly realistic EOS (e.g. IAPWS-IF95 standard for water) on unstructured curved hexahedra elements. The EOS covers the vapor, liquid-vapor, liquid and super critical regions. Inside the two-phase region thermodynamic equilibrium is assumed. For modeling the two phases a dense gas approach is used. The EOS is provided by the CoolProp library which can handle more than 100 fluids and we showed an efficient way to table this data to use it during CFD calculation on more than 6000 cores. The need to table the EOS was also shown because using the library directly during computation is too time consuming. The table approximates the EOS to a certain L_∞ error by using a quadtree approach with polynomial data representation and cut-cells for the saturation lines. Since real fluid effects like cavitation lead to high gradients in the conservative variables a shock capturing is implemented. During runtime the DG approximation of the solution can be replaced with a 2nd order finite volume scheme by refining the DG element into $(N + 1)^3$ FV elements. We showed that the Perrson indicator is well suited for 1D shock-tube calculation with phase change and for multi-dimensional calculations with cavitation. The DG-SEM is known for its excellent parallel efficiency and we showed that it still has a good scaling even with all these extensions.

Future work will concern the validation with 3D experimental data for fluid flows with cavitation. The code will be optimized to reduce the overhead of the table approach even more. To improve the scaling a load-balancing algorithm will be implemented to handle the load imbalances between the DG and FV method.

6 Acknowledgements

This work was supported by the Federal Ministry of Education and Research (BMBF) within the BMBF 3. HPC-Call project HONK “Industrialization of high-resolution numerical analysis of complex flow phenomena in hydraulic systems”. Computation resources were provided by the Gauss-Allianz. Our computations were carried out on the Hazel Hen supercomputer at the HLRS in Stuttgart.

References

- [1] C. Altmann, A. D. Beck, F. Hindenlang, M. Staudenmaier, G. J. Gassner, and C.-D. Munz. “An Efficient High Performance Parallelization of a Discontinuous Galerkin Spectral Element Method”. In: *Facing the Multicore-Challenge III: Aspects of New Paradigms and Technologies in Parallel Computing*.

- Ed. by R. Keller, D. Kramer, and J.-P. Weiss. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 37–47. ISBN: 978-3-642-35893-7. DOI: 10.1007/978-3-642-35893-7_4. URL: http://dx.doi.org/10.1007/978-3-642-35893-7_4.
- [2] F. Bassi and S. Rebay. “A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations”. In: *Journal of Computational Physics* 131.2 (1997), pp. 267–279. ISSN: 0021-9991. DOI: <http://dx.doi.org/10.1006/jcph.1996.5572>. URL: <http://www.sciencedirect.com/science/article/pii/S0021999196955722>.
 - [3] F. Bassi and S. Rebay. “Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier–Stokes equations”. In: *International journal for numerical methods in fluids* 40.1-2 (2002), pp. 197–207.
 - [4] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort. “Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp”. In: *Industrial & Engineering Chemistry Research* 53.6 (2014), pp. 2498–2508.
 - [5] C. E. Castro and E. F. Toro. “Roe-type Riemann solvers for general hyperbolic systems”. In: *International Journal for Numerical Methods in Fluids* 75.7 (2014), pp. 467–486.
 - [6] C. Chalons, F. Coquel, P. Engel, and C. Rohde. “Fast Relaxation Solvers for Hyperbolic-Elliptic Phase Transition Problems”. In: *SIAM Journal on Scientific Computing* 34.3 (June 2012). URL: <https://hal.archives-ouvertes.fr/hal-00726479>.
 - [7] B. Cockburn and C.-W. Shu. “Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems”. English. In: *Journal of Scientific Computing* 16.3 (2001), pp. 173–261.
 - [8] B. Cockburn, S. Hou, and C.-W. Shu. “The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case”. In: *Mathematics of Computation* 54.190 (1990), pp. 545–581.
 - [9] W. Dreyer, F. Duderstadt, M. Hantke, and G. Warnecke. “Bubbles in liquids with phase transition”. In: *Continuum Mechanics and Thermodynamics* 24.4 (2012), pp. 461–483. ISSN: 1432-0959. DOI: 10.1007/s00161-011-0225-6. URL: <http://dx.doi.org/10.1007/s00161-011-0225-6>.
 - [10] M. Dumbser, U. Iben, and C.-D. Munz. “Efficient implementation of high order unstructured WENO schemes for cavitating flows”. In: *Computers & Fluids* 86 (2013), pp. 141–168.
 - [11] S. Fechter, C.-D. Munz, C. Rohde, and C. Zeiler. “A sharp interface method for compressible liquid-vapor flow with phase transition and surface tension”. In: *arXiv preprint arXiv:1511.03612* (2015).
 - [12] S. F. Frisken and R. N. Perry. “Simple and efficient traversal methods for quadtrees and octrees”. In: *Journal of Graphics Tools* 7.3 (2002), pp. 1–11.
 - [13] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz. “Explicit discontinuous Galerkin methods for unsteady problems”. In: *Computers and Fluids* 61 (2012). ‘High Fidelity Flow Simulations’ Onera Scientific Day, pp. 86–93. ISSN: 0045-7930. DOI: <http://dx.doi.org/10.1016/j.compfluid.2012.03.006>. URL: <http://www.sciencedirect.com/science/article/pii/S004579301200093X>.
 - [14] P.-O. Persson and J. Peraire. “Sub-cell shock capturing for discontinuous Galerkin methods”. In: *AIAA paper* 112 (2006), p. 2006.
 - [15] W. Reed and T. Hill. “Triangular mesh methods for the neutron transport equation”. In: Oct. 1973. URL: <http://www.osti.gov/scitech/servlets/purl/4491151>.
 - [16] R. Saurel, F. Petitpas, and R. Abgrall. “Modelling phase transition in metastable liquids: application to cavitating and flashing flows”. In: *Journal of Fluid Mechanics* 607 (July 2008), pp. 313–350. ISSN: 1469-7645. DOI: 10.1017/S0022112008002061. URL: [http://journals.cambridge.org/article_S0022112008002061](http://journals.cambridge.org/article/S0022112008002061).
 - [17] R. Saurel, P. Cocchi, and P. B. Butler. “Numerical study of cavitation in the wake of a hypervelocity underwater projectile”. In: *Journal of Propulsion and power* 15.4 (1999), pp. 513–522.

- [18] R. Saurel, E. Franquet, E. Daniel, and O. Le Metayer. “A relaxation-projection method for compressible flows. Part I: The numerical equation of state for the Euler equations”. In: *Journal of Computational Physics* 223.2 (2007), pp. 822–845.
- [19] R. Span, W. Wagner, E. Lemmon, and R. Jacobsen. “Multiparameter equations of state - recent trends and future challenges”. In: *Fluid Phase Equilibria* 183-184 (2001). Proceedings of the fourteenth symposium on thermophysical properties, pp. 1–20.
- [20] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.
- [21] W. Wagner and A. Pruß. “The IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use”. In: *Journal of Physical and Chemical Reference Data* 31.2 (2002), pp. 387–535.
- [22] A. Zein, M. Hantke, and G. Warnecke. “Modeling Phase Transition for Compressible Two-phase Flows Applied to Metastable Liquids”. In: *J. Comput. Phys.* 229.8 (Apr. 2010), pp. 2964–2998. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2009.12.026. URL: <http://dx.doi.org/10.1016/j.jcp.2009.12.026>.